《C/C++学习指南》

第18.3讲：预处理指令 - define宏定义

作者：邵发　　QQ群：417024631
官网：http://www.afanihao.cn/
答疑：http://www.afanihao.cn/kbase/

预处理指令

以#开头的行，都预处理指令，用于指示编译器做一些预处理工作。比如#include "xxx.h"

这次课介绍 #define 指令，通过译为“宏定义”

注：预处理指令不是语句，行尾不要加分号

预处理指令

本篇介绍#define的两种用法

(1)定义一个“数值”
(2)定义一个“算式”

注：工程中应该避免使用这两种方式。

C/C++学习指南　邵发 www.afanihao.cn

#define用法1：定义一个"值"

源代码

```
#define PI 3.14
int main()
{
   double r = 1.2;
   double area = PI * r * r;
   return 0;
}
```

预处理之后的中间代码

```
int main()
{
    double r = 1.2;
    double area = 3.14 * r * r;
    return 0;
}
```

#define用法2：定义带参数的"算式"

源代码

```
#define  MAX(a,b)  a>b ? a : b
int main()
{
        int a = MAX(10, 12);
        return 0;
}
```

预处理之后的中间代码

```
int main()
{
    int a = 10 > 12 ? 10: 12; // 代入、替换、展开
    return 0;
}
```


#define常见的错误(1)

#define本身用处不大，容易出错，不建议使用。

例如，

```
#define  MUL  1 + 2
int main()
{
    int a = 4 * MUL; // a的值将是多少？？
    return 0;
}
```

记住：预处理过程是把#define的文本直接替换的，不会有任何计算动作。

#define的陷阱

记住一句话：#define代入的是文本，而不是值。

只要把文本原番不动地代入后，检查一下语法是否正确，程序是否合乎逻辑。

#define常见的错误(1)

所以，要定义一个值，通常外加一个小括号

```
#define  MUL  (1 + 2)
```

以免以外发生

#define常见的错误(2)

#define算式

```
#define  MAX(a,b)  a>b ? a : b
int main()
{
   int a=1,b=2;
   int r = MAX (a+b, a-b);
   return 0;
}
```


#define常见的错误(2)

所以，在算式中带参数时，一般将参数加括号

```
#define  MAX(a,b)  ((a)>(b) ? (a) : (b))
```

#define的取代办法

在程序中应该尽量少用这两种#define取代的办法是：

(1) 定义变量或const常量（第3章）

```
const double PI = 3.14;
```

(2) 定义inline函数（第8章）

```
inline int max(int a, int b)
{
    return  a>b ? a: b;
}
```


其他

(1)NULL的宏定义

(2)RAND_MAX的宏定义 （第16章，随机函数 rand()）